

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4065715号
(P4065715)

(45) 発行日 平成20年3月26日(2008.3.26)

(24) 登録日 平成20年1月11日(2008.1.11)

(51) Int.Cl. F 1
G 0 6 F 3 / 1 2 (2 0 0 6 . 0 1) G O 6 F 3 / 1 2 Z

請求項の数 6 (全 14 頁)

<p>(21) 出願番号 特願2002-120249 (P2002-120249) (22) 出願日 平成14年4月23日 (2002.4.23) (65) 公開番号 特開2003-316560 (P2003-316560A) (43) 公開日 平成15年11月7日 (2003.11.7) 審査請求日 平成17年4月19日 (2005.4.19)</p>	<p>(73) 特許権者 391005503 キヤノンアイテック株式会社 東京都府中市宮町1-40-1 (74) 代理人 100099748 弁理士 佐藤 克志 (72) 発明者 辻井 一誠 東京都府中市若松町1-38-1 東京電 子設計株式会社内 審査官 中田 剛史 (56) 参考文献 特開2000-118095 (JP, A)) 特開平02-040740 (JP, A)</p>
--	--

最終頁に続く

(54) 【発明の名称】 ジョブ処理装置

(57) 【特許請求の範囲】

【請求項1】

所定の記憶領域に格納された各ファイル进行处理するためのジョブが登録されたジョブリストを記憶する記憶手段と、

間欠的に、前記記憶領域に格納されている各ファイルについてのみ、当該ファイル进行处理するジョブが前記ジョブリストに登録されるように当該ジョブリストを更新するジョブリスト更新手段と、

前記ジョブリストに登録されたジョブを順次実行するジョブ実行手段とを有し、

前記ジョブリスト更新手段は、

ジョブリストに登録されたジョブに対応するファイルのファイル識別子を所定の基準に従ってソートして登録したジョブインデックスと、前記記憶領域に格納されたファイルのファイル識別子を前記基準に従ってソートして登録したカレントインデックスとを作成するソート手段と、

ジョブインデックスに登録されたファイル識別子を、ジョブインデックスにカレントインデックスに最後に登録されているファイル識別子と一致するファイル識別子が存在する場合には先頭のファイル識別子から少なくとも前記一致するファイル識別子まで、ジョブインデックスに登録された順番に、順次着目ファイル識別子として選択し、着目ファイル識別子を選択した各回について、カレントインデックスに登録されたファイル識別子のうち、今回に先行する回において着目ファイル識別子と一致したファイル識別子が存在する場合には今回に先行する直近の回において着目ファイル識別子と一致したファイル識別子よ

10

20

り後にカレントインデックスに登録されている各ファイル識別子を、他の場合にはカレントインデックスに登録されている各ファイル識別子を、少なくとも、今回の着目ファイル識別子と一致するファイル識別子がカレントインデックスの存在する場合には当該一致するファイル識別子まで、他の場合には最後のファイル識別子まで、カレントインデックスに登録された順番に順次検証ファイル識別子としながら、今回の着目ファイル識別子と各検証ファイル識別子とを比較し、当該着目ファイル識別子と検証ファイル識別子が一致した場合に、当該一致した検証ファイル識別子を除く他の今回検証ファイル識別子とした各ファイル識別子のファイル処理するジョブをジョブリストに追加し、当該着目ファイル識別子と一致する検証ファイル識別子が存在しなかった場合に、当該着目ファイル識別子としたファイル識別子のファイルのジョブをジョブリストから削除するジョブ登録処理手段を有することを特徴とするジョブ処理装置。

10

【請求項 2】

請求項 1 記載のジョブ処理装置であって、

前記ジョブ登録処理手段は、

ジョブインデックスに最後に登録されたファイル識別子を着目ファイル識別子として選択した回について、当該着目ファイル識別子と検証ファイル識別子が一致し、かつ、当該一致した検証ファイル識別子としたファイル識別子より後にカレントインデックスに登録されているファイル識別子が存在する場合には、当該存在するファイル識別子のファイル処理するジョブをジョブリストに追加し、当該着目ファイル識別子と一致する検証ファイル識別子が存在しなかった場合に、当該着目ファイル識別子としたファイル識別子のファイルのジョブをジョブリストから削除し、今回検証ファイル識別子とした各ファイル識別子のファイル処理するジョブをジョブリストに追加することを特徴とするジョブ処理装置。

20

【請求項 3】

請求項 1 または 2 記載のジョブ処理装置であって、

前記ジョブ登録処理手段は、

カレントインデックスに最後に登録されたファイル識別子である検証ファイル識別子が着目ファイル識別子と一致し、当該着目ファイル識別子として選択されたファイル識別子より後にジョブインデックスに登録されたファイル識別子が存在する場合には、当該存在するファイル識別子のファイル処理するジョブをジョブリストから削除し、当該着目ファイル識別子として選択されたファイル識別子より後にジョブインデックスに登録されたファイル識別子を着目ファイル識別子として選択しないことを特徴とするジョブ処理装置。

30

【請求項 4】

請求項 1、2 または 3 記載のジョブ処理装置であって、

前記記憶領域はフォルダであって、前記ファイル識別子はファイル名であって、前記ジョブは前記ファイルの処理としてファイルを印刷処理を行うものであることを特徴とするジョブ処理装置。

【請求項 5】

コンピュータシステムにおいてジョブリストにジョブを登録し、前記ジョブリストに登録されたジョブを実行するジョブ処理方法であって、

40

間欠的に、所定の条件を満たす各データについてのみ、当該データを処理するジョブがジョブリストに登録されるように当該ジョブリストを更新するジョブリスト更新ステップと、

前記ジョブリストに登録されたジョブを順次実行するジョブ実行ステップとを有し、

前記ジョブリスト更新ステップは、

ジョブリストに登録されたジョブに対応するデータのデータ識別子を所定の基準に従ってソートして登録したジョブインデックスと、前記記憶領域に格納されたデータのデータ識別子を前記基準に従ってソートして登録したカレントインデックスとを作成するステップと、

ジョブインデックスに登録された各データ識別子を、ジョブインデックスにカレントイン

50

デックスに最後に登録されているデータ識別子と一致するデータ識別子が存在する場合には先頭のデータ識別子から少なくとも前記一致するデータ識別子まで、ジョブインデックスに登録された順番に、順次着目データ識別子として選択し、着目データ識別子を選択した各回について、カレントインデックスに登録されたデータ識別子のうち、今回に先行する回において着目データ識別子と一致したデータ識別子が存在する場合には今回に先行する直近の回において着目データ識別子と一致したデータ識別子より後にカレントインデックスに登録されているデータ識別子を、他の場合にはカレントインデックスに登録されている各データ識別子を、少なくとも、今回の着目データ識別子と一致するデータ識別子がカレントインデックスの存在する場合には当該一致するデータ識別子まで、他の場合には最後のデータ識別子まで、カレントインデックスに登録された順番に順次検証データ識別子としながら、今回の着目データ識別子と検証データ識別子とを比較し、当該着目データ識別子と検証データ識別子が一致した場合に、当該一致した検証データ識別子を除く他の今回検証データ識別子とした各データ識別子のデータを処理するジョブをジョブリストに追加し、当該着目データ識別子と一致する検証データ識別子が存在しなかった場合に、当該着目データ識別子としたデータ識別子のデータのジョブをジョブリストから削除するステップとを有することを特徴とするジョブ処理方法。

10

【請求項6】

コンピュータによって読み取られ実行されるコンピュータプログラムであって、前記コンピュータプログラムは、前記コンピュータに間欠的に、前記コンピュータ上の所定の条件を満たす各データについてのみ、当該データを処理するジョブがジョブリストに登録されるように当該ジョブリストを更新するジョブリスト更新ステップと、前記ジョブリストに登録されたジョブを順次実行するジョブ実行ステップとを実行させ、前記ジョブリスト更新ステップは、ジョブリストに登録されたジョブに対応するデータのデータ識別子を所定の基準に従ってソートして登録したジョブインデックスと、前記記憶領域に格納されたデータのデータ識別子を前記基準に従ってソートして登録したカレントインデックスとを作成するステップと、ジョブインデックスに登録された各データ識別子を、ジョブインデックスにカレントインデックスに最後に登録されているデータ識別子と一致するデータ識別子が存在する場合には先頭のデータ識別子から少なくとも前記一致するデータ識別子まで、ジョブインデックスに登録された順番に、順次着目データ識別子として選択し、着目データ識別子を選択した各回について、カレントインデックスに登録されたデータ識別子のうち、今回に先行する回において着目データ識別子と一致したデータ識別子が存在する場合には今回に先行する直近の回において着目データ識別子と一致したデータ識別子より後にカレントインデックスに登録されているデータ識別子を、他の場合にはカレントインデックスに登録されている各データ識別子を、少なくとも、今回の着目データ識別子と一致するデータ識別子がカレントインデックスの存在する場合には当該一致するデータ識別子まで、他の場合には最後のデータ識別子まで、カレントインデックスに登録された順番に順次検証データ識別子としながら、今回の着目データ識別子と各検証データ識別子とを比較し、当該着目データ識別子と検証データ識別子が一致した場合に、当該一致した検証データ識別子を除く他の今回検証データ識別子とした各データ識別子のデータを処理するジョブをジョブリストに追加し、当該着目データ識別子と一致する検証データ識別子が存在しなかった場合に、当該着目データ識別子としたデータ識別子のデータのジョブをジョブリストから削除するステップとを有することを特徴とするコンピュータプログラム。

20

30

40

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、コンピュータシステム等においてジョブを処理する技術に関し、特に、ホットフォルダに格納されたドキュメントファイルの印刷をジョブとして登録し、これを処理す

50

る印刷制御システムに関するものである。

【0002】

【従来の技術】

コンピュータシステム等においてジョブを処理する技術としては、たとえば、特開2000-118095号公報に記載された、ホットフォルダに格納されたファイルの印刷をジョブとして登録し、これを処理する技術が知られている。

【0003】

この技術は、定期的にホットフォルダへのドキュメントファイルの格納を監視し、新たにホットフォルダに格納されたドキュメントファイルが検出されたならば、検出されたドキュメントファイルの印刷ジョブをジョブリストに登録すると共に、ジョブリストに登録された印刷ジョブを逐次実行することにより、ホットフォルダに格納された各ドキュメントファイルの印刷を実現するものである。

10

【0004】

このような技術によれば、ユーザは、ドキュメントファイルを印刷したいときに、そのドキュメントファイルをホットフォルダへ移動/格納するだけで、その印刷を行うことができるようになる。

【0005】

【発明が解決しようとする課題】

前記従来の技術は、定期的に、ホットフォルダに新規に格納されたドキュメントファイルの印刷ジョブを登録するものである。したがって、印刷ジョブの登録のためには、ホットフォルダに格納されたドキュメントファイルのうち、どれが新規にホットフォルダに格納されたドキュメントファイルであり、どれが既に印刷ジョブを登録したドキュメントファイルであるのかの識別を行う必要がある。また、印刷ジョブの実行前にホットフォルダから消去/移動されたドキュメントファイルについては、そのドキュメントファイルについての印刷ジョブの登録をキャンセルする必要がある。

20

【0006】

そして、このために、ホットフォルダに格納されたドキュメントファイルの数が増大するに従い、定期的に行われる印刷ジョブ登録の処理の処理量が増大し、システムリソースを多く消費してしまうという問題が生じる。

そこで、本発明は、ジョブの対象となるデータの存在に基づいて行うジョブ登録の処理を効率化することを課題とする。

30

【0007】

【課題を解決するための手段】

前記課題達成のために、本発明は、前記課題達成のために、たとえば、ジョブリストにジョブを登録し、前記ジョブリストに登録されたジョブを実行するジョブ処理方法であって、

間欠的に、所定の条件を満たす各データについてのみ、当該データを処理するジョブがジョブリストに登録されるように当該ジョブリストを更新するジョブリスト更新ステップと

、前記ジョブリストに登録されたジョブを順次実行するジョブ実行ステップとを有し、前記ジョブリスト更新ステップは、

40

ジョブリストに登録されたジョブに対応するデータのデータ識別子を所定の基準に従ってソートして登録したジョブインデックスと、前記記憶領域に格納されたデータのデータ識別子を前記基準に従ってソートして登録したカレントインデックスとを作成するステップと、

ジョブインデックスに登録された各データ識別子を、ジョブインデックスにカレントインデックスに最後に登録されているデータ識別子と一致するデータ識別子が存在する場合には先頭のデータ識別子から少なくとも前記一致するデータ識別子まで、ジョブインデックスに登録された順番に、順次着目データ識別子として選択し、着目データ識別子を選択した各回について、カレントインデックスに登録されたデータ識別子のうち、今回に先行す

50

る回において着目データ識別子と一致したデータ識別子が存在する場合には今回に先行する直近の回において着目データ識別子と一致したデータ識別子より後にカレントインデックスに登録されているデータ識別子を、他の場合にはカレントインデックスに登録されている各データ識別子を、少なくとも、今回の着目データ識別子と一致するデータ識別子がカレントインデックスの存在する場合には当該一致するデータ識別子まで、他の場合には最後のデータ識別子まで、カレントインデックスに登録された順番に順次検証データ識別子としながら、今回の着目データ識別子と検証データ識別子とを比較し、当該着目データ識別子と検証データ識別子が一致した場合に、当該一致した検証データ識別子を除く他の今回検証データ識別子とした各データ識別子のデータを処理するジョブをジョブリストに追加し、当該着目データ識別子と一致する検証データ識別子が存在しなかった場合に、当該着目データ識別子としたデータ識別子のデータのジョブをジョブリストから削除するステップとを有することを特徴とするジョブ処理方法を提供する。

10

【0008】

このようなジョブ処理方法によれば、以下の実施形態において、その一例を詳細に説明するように、ジョブに対応するファイルが前記記憶領域から削除されていないかどうかを調べるために、ジョブリストに登録されたジョブに対応するファイルのファイル識別子のそれぞれについて、前記記憶領域に格納されたファイルのファイル識別子のそれぞれと比較したり、新たにジョブを生成すべきファイルが前記記憶領域に追加されていないかどうかを調べるために、前記記憶領域に格納されたファイルのファイル識別子のそれぞれについて、ジョブリストに登録された各ジョブに対応するファイルのファイル識別子のそれぞれと比較する場合に比べ、より少ない比較回数で、前記記憶領域から削除されたファイルに対応するジョブのファイル識別子や前記記憶領域に新たに追加されたファイルのファイル識別子を算出することができるようになる。

20

【0009】**【発明の実施の形態】**

以下、本発明の実施形態について説明する。

図1に、本実施形態に係るコンピュータシステムの構成を示す。

図示するように本コンピュータシステムは、コンピュータ1とプリンタ2が接続されたシステムであり、コンピュータ1の上ではプリンタ2の印刷動作を制御するプリンタドライバ111が組み込まれたオペレーティングシステム11と、複数のアプリケーション12

30

～14が稼働している。

【0010】

また、これらアプリケーションのうち、アプリケーション12は、ドキュメントファイルの印刷処理を行う印刷処理サーバであり、ジョブ実行制御部121とジョブ監視部122とを有する。また、他のアプリケーションはたとえば、ドキュメントファイルを編集、生成するアプリケーションなどである。

【0011】

ここで、コンピュータ1は、ハードウェア的には、CPUや、主記憶や、ハードディスク装置等の外部記憶装置や、キーボードやポインティングデバイスなどの入力装置や、表示装置などを備えた、一般的な構成を有するコンピュータ1であり。この場合、コンピュータ1の以上の各部は、CPUが主記憶にロードされたプログラムを実行することによりコンピュータに具現化されるとして実現される。また、以上のコンピュータ1の機能は、ネットワーク上に分散配置されたものであってもかまわない。

40

【0012】

以下、このようなコンピュータシステムにおける印刷処理サーバ12の動作について説明する。

図2に示すように、印刷処理サーバ12は、オペレーティングシステム11のファイルシステムが管理するディレクトリの一種であるホットフォルダに格納されたドキュメントファイルを印刷を行うものである。

この印刷動作は、大きく分けて、ジョブ監視部122によって定期的に行われる、ホット

50

フォルダのドキュメントファイル格納状態に応じたジョブリストの登録ジョブの更新と、ジョブ実行制御部 1 2 1 によるジョブリストの登録ジョブの逐次実行とよりなる。

【 0 0 1 3 】

ここで、ジョブ監視部 1 2 2 によってジョブリストに登録される各ジョブは、ホットフォルダ内の一つのドキュメントファイルの印刷を要求するものであり、ジョブ実行制御部 1 2 1 によるジョブリストの登録ジョブの実行とは、そのジョブが要求するドキュメントファイルのプリンタ 2 による印刷を、オペレーティングシステム 1 1 とプリンタドライバ 1 1 1 を介して制御するものである。

【 0 0 1 4 】

さて、ジョブリストには、各ジョブの情報として、当該ジョブが要求するドキュメントファイルのファイル名 (F i l e n a m e) と当該ジョブがジョブリストに登録された日時 (D a t e) との組が登録される。一方、ジョブ監視部 1 2 2 は、このようなジョブリストの登録ジョブの更新を、登録ファイルインデックスとカレントファイルインデックスと呼ぶ、二つの変数テーブルを用いながら行う。そして、登録ファイルインデックスは、ファイル名 (F i l e n a m e) と削除フラグ (d e l e t e) の組が記述されるエントリを複数もつことができ、カレントファイルインデックスはファイル名 (F i l e n a m e) と新規フラグ (n e w) の組が記述されるエントリを複数もつことができる。

10

【 0 0 1 5 】

以下、このジョブ監視部 1 2 2 の動作について説明する。

図 3 に、ジョブ監視部 1 2 2 の行うジョブリスト更新処理の手順を示す。

20

図示するように、ジョブ監視部 1 2 2 は、起動されると、まず、登録ファイルインデックスを生成する (ステップ 3 0 2) 。すなわち、ジョブリストに格納されているファイル名を、ファイル名をソートキーとしてソートし、各ファイル名を、ソート後の順番に従って、登録ファイルインデックスに、そのファイル名を記述するエントリを作成して登録する。ファイル名のソートは、たとえば、ファイル名を、アルファベット順や文字コード順に並べるなどの手法などにより行う。ただし、このようなソートキーは、ある一定のルールにより一意にファイル名に順番づけを行えるものであれば任意のキーによって良い。また、ここでは、登録ファイルインデックスに登録した各ファイル名のエントリの削除フラグは、全てオフとする。

【 0 0 1 6 】

30

次に、ジョブ監視部 1 2 2 は、カレントファイルインデックスを作成する (ステップ 3 0 4) 。すなわち、ホットフォルダに格納されているファイル名をオペレーティングシステム 1 1 を介して入手し、入手したファイル名を、ファイル名をソートキーとしてソートし、各ファイル名を、ソート後の順番に従って、カレントファイルインデックスに、そのファイル名を記述するエントリを作成して登録する。ファイル名のソートは、前記登録ファイルインデックスと同じルールにより行う。また、ここでは、カレントファイルインデックスに登録した各ファイル名のエントリの新規フラグは、全てオフとする。

【 0 0 1 7 】

次に、登録ファイルインデックスに登録されたファイル名の数を $M A X j$ 、カレントファイルインデックスに登録されたファイル名の数を $M A X c$ とする (ステップ 3 0 6) 。そして、着目登録ファイル番号 j を 1 に、比較ファイル番号 c を 1 に、比較回数 t を 0 に初期化する (ステップ 3 0 8) 。

40

【 0 0 1 8 】

そして、登録ファイルインデックスの j 番目のファイル名とカレントファイルインデックスの c 番目のファイル名の比較を (ステップ 3 1 0) 、両者が一致するか (ステップ 3 1 2) 、両者が一致しないままカレントファイルインデックスの最後 ($M A X c$ 番目) のファイル名との比較が終了するまで (ステップ 3 3 2) 、比較ファイル番号 c と、比較回数 t を 1 増加しながら (ステップ 3 4 0) 行う。

【 0 0 1 9 】

そして、もし、両者が一致しないまま、カレントファイルインデックスの最後 ($M A X c$

50

番目)のファイル名との比較が終了したならば(ステップ332)、登録ファイルインデックスのj番目のファイル名のエントリの削除フラグをオンに設定する(ステップ334)。そして、着目登録ファイル番号jが、登録ファイルインデックスのファイル名数MAXjと一致するかどうかを調べ(ステップ336)、そうでなければ、着目登録ファイル番号jを1増加し、比較ファイル番号cをC-tとし、tを0とし(ステップ342)、ステップ310に戻って、登録ファイルインデックスに次に登録されたファイル名の処理を行う。

【0020】

一方、着目登録ファイル番号jが、登録ファイルインデックスのファイル名数MAXjと一致する場合には(ステップ336)、カレントファイルインデックスのC-t番目からMAXc番目までのファイル名のエントリの新規フラグをオンに設定し(ステップ338)、ステップ324に進む。

10

【0021】

さて、ステップ312において、登録ファイルインデックスのj番目のファイル名とカレントファイルインデックスのc番目のファイル名が一致した場合には、次のように処理を行う。

すなわち、比較回数tが0かどうかを調べ(ステップ314)、0でなければ、カレントファイルインデックスのc-t番目からc-1番目までのファイル名のエントリの新規フラグをオンに設定し(ステップ316)、ステップ318に進む。一方、比較回数tが0であれば、そのままステップ318に進む。

20

【0022】

ステップ318では、比較ファイル番号cがMAXcと一致するかどうかを調べ、一致すれば、登録ファイルインデックスのj+1番目からMAXj番目のファイル名のエントリの削除フラグをオンに設定し(ステップ344)、ステップ324に進む。一方、比較ファイル番号cがMAXcと一致しない場合には(ステップ318)、さらに着目登録ファイル番号jがMAXjと一致するかどうかを調べ(ステップ320)、一致する場合には、カレントファイルインデックスのc+1番目からMAXc番目のファイル名のエントリの新規フラグをオンに設定し(ステップ322)、ステップ324に進む。そして、ステップ320で、着目登録ファイル番号jがMAXjと一致しなかった場合には、着目登録ファイル番号jと比較ファイル番号cを1増加し、tを0とし(ステップ346)、ステップ310に戻って、登録ファイルインデックスに次に登録されたファイル名の処理を行う。

30

【0023】

さて、ステップ324では、登録ファイルインデックスの削除フラグがオンに設定されているエントリのファイル名と一致するファイル名のエントリをジョブリストから削除する。そして、ステップ326において、カレントファイルインデックスの新規フラグがオンに設定されているエントリのファイル名を、順次、ジョブリストにエントリを作成して追加し、各エントリの日時に現在日時を設定する。

【0024】

そして、タイマを設定し(ステップ328)、タイマがタイムアウトしたならば(ステップ330)、以上の処理を再度行う。すなわち、ステップ302~ステップ326の処理は定期的に行われる。

40

ここで、以上の処理によって、どのようにホットフォルダのドキュメントファイルの格納状態に応じてジョブリストの内容が更新されていくかを、その例によって説明する。

いま、図4aに示すように、ステップ302で、その時点のジョブリストの内容に基づき登録ファイルインデックスにE、F、Jのファイル名のエントリが当該順番で作成され、ステップ304で、その時点のホットフォルダに格納されたドキュメントファイルに基づき、カレントファイルインデックスにA~E、G~Kのファイル名のエントリが当該順番で作成された場合を例にとる。

【0025】

50

すなわち、この例は、前回のジョブリストの更新後に、ドキュメントファイルFがホットフォルダから削除され、かつ、前回のジョブリストの更新後に、A～D、G～I、Kのドキュメントファイルがホットフォルダに追加された場合についてのものである。

【0026】

この場合、まず、登録ファイルインデックスの最初のファイル名Eが $j = 1$ のファイル名として選択され、ステップ310で、カレントファイルインデックスの先頭($c = 1$)から登録順に(ステップ340)、カレントファイルインデックスに登録されたファイル名と比較されていく。

【0027】

そして、 $c = 5$ 、 $t = 4$ となった時点で、登録ファイルインデックスの最初のファイル名Eと、カレントファイルインデックスの5番目のファイル名Eが一致する(ステップ312)ので、ステップ316で、カレントファイルインデックスの $c - t = 5 - 4 = 1$ 番目のファイル名Aのエントリから $c - 1 = 5 - 1 = 4$ 番目のファイル名Dまでのエントリの新規フラグがオンに設定される。

【0028】

すなわち、以上の処理では、カレントファイルインデックスに登録されたファイル名のうち、登録ファイルインデックスに登録された最初のファイル名と一致するファイル名より、前のファイル名は全て新規にホットフォルダに格納されたドキュメントフォルダのものであるので、その新規フラグをオンに設定する。

【0029】

なお、比較回数 t は、登録ファイルインデックスの j 番目のファイル名につき行われた、ファイル名が一致しなかった比較の回数を表すようになっていく。比較ファイル番号 c は、登録ファイルインデックスの j 番目のファイル名と比較するファイル名のカレントファイルインデックス中の順番を表すようになっていく。

【0030】

次に、ステップ346で j が1増加されて2、 c が1増加されて6、 $t = 0$ とされて、登録ファイルインデックスの2番目のファイル名Fが $j = 2$ のファイル名として選択され、カレントファイルインデックスの $c (= 6)$ 番目のファイル名Gから登録順に(ステップ340)、カレントファイルインデックスに登録されたファイル名と比較されていく。なお、ここで c と j を1増加するのは、増加前の c と j について、登録ファイルインデックスの先頭から j 番目のファイル名まで対応するドキュメントファイルがホットフォルダから削除されていないかが判定済みであり、カレントファイルインデックスの先頭から c 番目のファイル名まで対応するドキュメントファイルがホットフォルダに新規に追加されたものである判定済みであり、以降の処理では、これら判定済みのファイル名は処理の対象から除外するためである。

【0031】

さて、 $j = 2$ では、登録ファイルインデックスに登録されたファイル名Fのドキュメントファイルが既にホットフォルダから削除されているので、ステップ346の比較が一致することはなく、カレントファイルインデックスの $MAX c (= 10)$ 番目のファイル名Kとの比較が終了する。この時点で、ファイル名が一致しなかった比較回数を表す比較回数は $t = 4$ となっている。すると、ステップ332の判定に従って、ステップ334へ処理が分岐され、登録ファイルインデックスの $j (= 2)$ 番目のファイル名Fのエントリの削除フラグがオンに設定される。

【0032】

すなわち、登録ファイルインデックスに登録されたファイル名のうち、カレントファイルインデックスに登録された最後のファイル名まで比較しても一致するファイル名が存在しないファイル名は、ホットフォルダから削除された対応するドキュメントファイルのものであるので、削除フラグをオンに設定する。

【0033】

そして、ステップ342で、 j が1増加されて3、 c が $c - t = 10 - 4 = 5$ とされて6

10

20

30

40

50

、 $t = 0$ とされて、登録ファイルインデックスの3番目のファイル名Jが $j = 2$ のファイル名として選択され、カレントファイルインデックスの $c (= 6)$ 番目のファイル名Gから登録順に(ステップ340)、カレントファイルインデックスに登録されたファイル名と比較されていく。

【0034】

ここで、先程と異なり、 c を $c - t$ とするのは、更新前の j と c について、登録ファイルインデックスの j 番目のファイル名と一致するファイル名がカレントファイルインデックスに登録されていなかったために、登録ファイルインデックスの j 番目のファイル名との比較を開始した $c - t$ 番目のファイル名から c 番目のファイル名まで、それが新規ホットフォルダに追加されたドキュメントファイルであるかどうか判定できなかったためである。

10

【0035】

さて、登録ファイルインデックスの3番目のファイル名Jは、 $c = 9$ 、 $t = 3$ となった時点で、カレントファイルインデックスに9番目に登録されたファイル名Jと一致する。そこで、ステップ316で、カレントファイルインデックスの $c - t = 9 - 3 = 6$ 番目のファイル名Gのエントリから $c - 1 = 9 - 1 = 8$ 番目のファイル名Iまでのエントリの新規フラグがオンに設定される。

【0036】

すなわち、カレントファイルインデックスに登録されたファイル名のうち、カレントファイルインデックスのファイル名のうち、登録ファイルインデックスに登録された j 番目のファイル名との比較を開始したファイル名から、当該 j 番目のファイル名と一致するファイル名の前のファイル名までの間のファイル名は全て新規にホットフォルダに格納されたドキュメントフォルダのものであるので、その新規フラグをオンに設定する。

20

【0037】

次に、この時、 $j = \text{MAX } j = 3$ となるので、ステップ320により、処理はステップ322に分岐され、カレントファイルインデックスの $c + 1 = 10$ 番目のファイル名Kのエントリ $\sim \text{MAX } c = 10$ 番目のファイル名Kのエントリまでのエントリの新規フラグがオンに設定され、ステップ324へ処理が進む。すなわち、カレントファイルインデックスのファイル名のうち、登録ファイルの最後のファイル名と一致したファイル名より後のファイル名は、全て新規にホットフォルダに格納されたドキュメントフォルダのものであるので、その新規フラグをオンに設定する。

30

【0038】

さて、この時点では、以上の処理により登録ファイルインデックスのファイル名Fのエントリの削除フラグがオンに設定され、カレントファイルインデックスのファイル名A \sim D、G \sim I、Kのエントリの新規フラグがオンに設定されている。

【0039】

したがって、ステップ324ではジョブリストのファイル名Fのジョブが削除され、ステップ326ではファイル名A \sim D、G \sim I、Kのジョブがジョブリストに追加されることになる。

次に、図4bに、他の処理例を示す。

40

この例では、ステップ302で、その時点のジョブリストの内容に基づき登録ファイルインデックスにE、H、J、Lのファイル名のエントリが当該順番で作成され、ステップ304で、その時点のホットフォルダに格納されたドキュメントファイルに基づき、カレントファイルインデックスにA \sim E、G、Hのファイル名のエントリが当該順番で作成された場合を例にとる。

【0040】

すなわち、この例は、前回のジョブリストの更新後に、ドキュメントファイルJとLがホットフォルダから削除され、かつ、前回のジョブリストの更新後に、A \sim D、Gのドキュメントファイルがホットフォルダに追加された場合についてのものである。

【0041】

50

この場合、図4 aの例の場合と同様に、まず、登録ファイルインデックスの最初のファイル名Eが $j = 1$ のファイル名として選択され、ステップ3 1 0で、カレントファイルインデックスの先頭 ($c = 1$) から登録順に (ステップ3 4 0)、カレントファイルインデックスに登録されたファイル名と比較されていく。

【0042】

そして、 $c = 5$ 、 $t = 4$ となった時点で、登録ファイルインデックスの最初のファイル名Eと、カレントファイルインデックスの5番目のファイル名Eが一致する (ステップ3 1 2) ので、ステップ3 1 6で、カレントファイルインデックスの $c - t = 5 - 4 = 1$ 番目のファイル名Aのエントリから $c - 1 = 5 - 1 = 4$ 番目のファイル名Dまでのエントリの新規フラグがオンに設定される。

10

【0043】

次に、ステップ3 4 6で j が1増加されて2、 c が1増加されて6、 $t = 0$ とされて、登録ファイルインデックスの2番目のファイル名Hが $j = 2$ のファイル名として選択され、カレントファイルインデックスの $c (= 6)$ 番目のファイル名Gから登録順に (ステップ3 4 0)、カレントファイルインデックスに登録されたファイル名と比較されていく。

【0044】

そして、 $c = 7$ 、 $t = 1$ となった時点で、カレントファイルインデックスに7番目に登録されたファイル名Hと一致する。そこで、ステップ3 1 6で、カレントファイルインデックスの $c - t = 7 - 1 = 6$ 番目のファイル名Gのエントリから $c - 1 = 7 - 1 = 6$ 番目のファイル名Gまでのエントリの新規フラグがオンに設定される。

20

【0045】

次に、この時、 $c = \text{MAX } c = 7$ となるので、ステップ3 1 8により、処理はステップ3 4 4に分岐され、登録ファイルインデックスの $j + 1 = 3$ 番目のファイル名Jのエントリ ~ $\text{MAX } j = 4$ 番目のファイル名Lエントリまでのエントリの削除フラグがオンに設定され、ステップ3 2 4へ処理が進む。

【0046】

すなわち、登録ファイルインデックスに登録されたファイル名のうち、カレントファイルインデックスの最後のファイル名と一致するファイル名より後に登録されたファイル名は、全てホットフォルダから削除されたドキュメントファイルのものであるので、削除フラグをオンに設定する。

30

【0047】

さて、この時点では、以上の処理により登録ファイルインデックスのファイル名J、Lのエントリの削除フラグがオンに設定され、カレントファイルインデックスのファイル名A ~ D、Gのエントリの新規フラグがオンに設定されている。したがって、ステップ3 2 4ではジョブリストのファイル名J、Lのジョブが削除され、ステップ3 2 6ではファイル名A ~ D、Gのジョブがジョブリストに追加されることになる。

次に、図4 cに他の処理例を示す。

すなわち、この例は、前回のジョブリストの更新後に、ドキュメントファイルHがホットフォルダから削除され、かつ、前回のジョブリストの更新後に、A ~ D、G、J、Kのドキュメントファイルがホットフォルダに追加された場合についてのものである。

40

【0048】

この場合、図4 aの例の場合と同様に、まず、登録ファイルインデックスの最初のファイル名Eが $j = 1$ のファイル名として選択され、ステップ3 1 0で、カレントファイルインデックスの先頭 ($c = 1$) から登録順に (ステップ3 4 0)、カレントファイルインデックスに登録されたファイル名と比較されていく。

【0049】

そして、 $c = 5$ 、 $t = 4$ となった時点で、登録ファイルインデックスの最初のファイル名Eと、カレントファイルインデックスの5番目のファイル名Eが一致する (ステップ3 1 2) ので、ステップ3 1 6で、カレントファイルインデックスの $c - t = 5 - 4 = 1$ 番目のファイル名Aのエントリから $c - 1 = 5 - 1 = 4$ 番目のファイル名Dまでのエントリの

50

新規フラグがオンに設定される。

【0050】

そして、次に、ステップ346でjが1増加されて2、cが1増加されて6、t=0とされて、登録ファイルインデックスの2番目のファイル名Hがj=2のファイル名として選択され、カレントファイルインデックスのc(=6)番目のファイル名Gから登録順に(ステップ340)、カレントファイルインデックスに登録されたファイル名と比較されていく。

【0051】

今回は、登録ファイルインデックスに登録されたファイル名Hのドキュメントファイルが既にホットフォルダから削除されているので、ステップ346の比較が一致することはない、カレントファイルインデックスのMAXc(=8)番目のファイル名Kとの比較が終了する。この時点で、ファイル名が一致しなかった比較回数を表す比較回数はt=2となっている。すると、ステップ332の判定に従って、ステップ334へ処理が分岐され、登録ファイルインデックスのj(=2)番目のファイル名Hのエントリの削除フラグがオンに設定される。また、この場合は、j=MAXj=2となっているので、ステップ336の判定に従ってステップ338に処理が分岐され、カレントファイルのc-t=8-2=6番目のファイル名GのエントリからMAXc=8番目のファイル名Kのエントリまでの各エントリの新規フラグがオンに設定され、ステップ324に処理が進む。

【0052】

すなわち、登録ファイルインデックスに登録された最後のファイル名が、カレントファイルインデックスの最後のファイル名と一致しなかった場合には、カレントファイルインデックスのファイル名のうち、登録ファイルインデックスに登録された最後のファイル名との比較を開始したファイル名より以降のファイル名は、全てホットフォルダに新規に格納されたドキュメントファイルのものであるので、新規フラグを設定する。

【0053】

さて、この時点では、以上の処理により登録ファイルインデックスのファイル名Hのエントリの削除フラグがオンに設定され、カレントファイルインデックスのファイル名A~D、G~Kのエントリの新規フラグがオンに設定されている。したがって、ステップ324ではジョブリストのファイル名Hのジョブが削除され、ステップ326ではファイル名A~D、G~Kのジョブがジョブリストに追加されることになる。

以上、ジョブ監視部122の動作について説明した。

次に、ジョブ実行制御部121の動作について説明する。

ジョブ実行制御部121は、処理を開始すると、図5に示すジョブ実行処理を行う。

このジョブ実行処理では、図示するようにジョブリストにジョブが登録されているかどうかを監視し(ステップ502)、ジョブが登録されていれば、ジョブリストの先頭のジョブを取り出し、取り出したジョブのエントリに記述されているドキュメントファイルをホットフォルダから読み出して(ステップ504)、オペレーティングシステム11/プリンタドライバ111を介して、ドキュメントフォルダのプリンタ2における印刷を制御する(ステップ506)。

【0054】

そして、印刷が正常に終了したならば(ステップ508)、印刷を終了したジョブをジョブリストから削除し(ステップ510)、オペレーティングシステム11に要請し、印刷したドキュメントファイルを、ホットフォルダから予め用意された処理済みフォルダに移動する(ステップ512)。

【0055】

以上、本発明の実施形態について説明した。

ところで、以上では、ドキュメントファイルをジョブで処理する対象として、ジョブをドキュメントファイルを印刷するジョブとして説明したが、以上のジョブリストの更新の処理は、任意のキーによりソートすることができるデータを対象として、任意の処理を行うジョブについて同様に適用可能である。たとえば、メールフォルダに格納されたメールデ

10

20

30

40

50

ータを対象として、メール送信を行うジョブや、その他のジョブのジョブリストの更新に同様に適用可能である。また、ジョブの処理の対象とするデータは、必ずしも一つのフォルダに格納されたものとする必要はなく、任意の条件で選定されるデータをジョブの処理の対象とするようにして良い。

【 0 0 5 6 】

【 発 明 の 効 果 】

以上のように、本発明によれば、ジョブの対象となるデータの存在に基づいて行うジョブ登録の処理を効率化することができる。

【 図 面 の 簡 単 な 説 明 】

【 図 1 】 本発明の実施形態に係るコンピュータシステムの構成を示すブロック図である。

10

【 図 2 】 本発明の実施形態に係る印刷処理サーバ12とホットフォルダの関係を示す図である。

【 図 3 】 本発明の実施形態に係るジョブリスト更新処理の手順を示すフローチャートである。

【 図 4 】 本発明の実施形態に係るジョブリスト更新処理の具体例を示す図である。

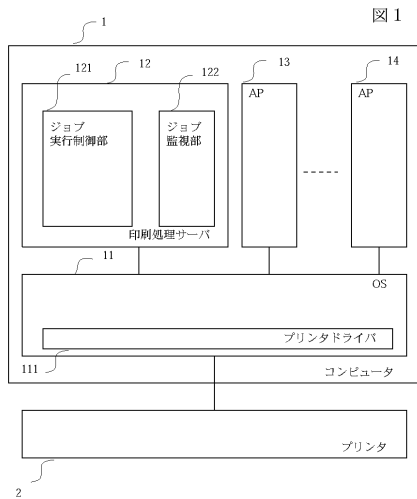
【 図 5 】 本発明の実施形態に係るジョブ実行処理の手順を示すフローチャートである。

【 符 号 の 説 明 】

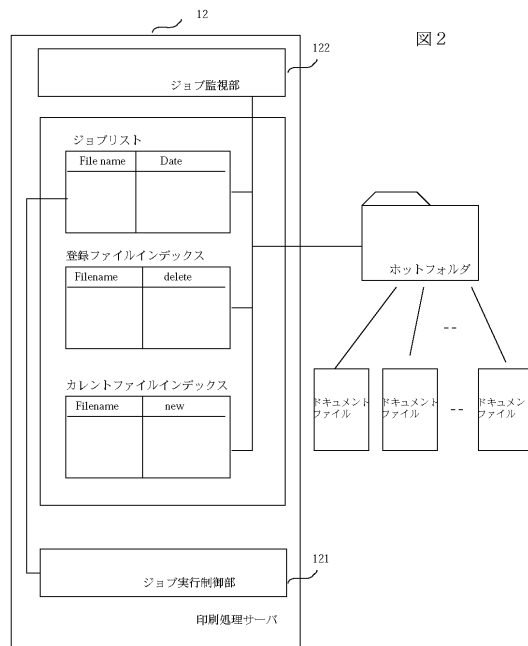
1：コンピュータ、2：プリンタ、11：オペレーティングシステム、12：印刷処理サーバ、13～14：アプリケーション、111：プリンタドライバ、121：ジョブ実行制御部、122：ジョブ監視部。

20

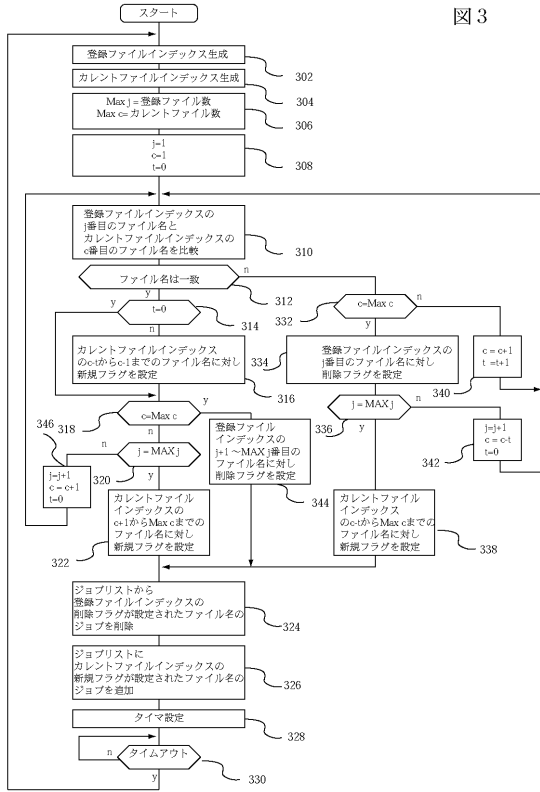
【 図 1 】



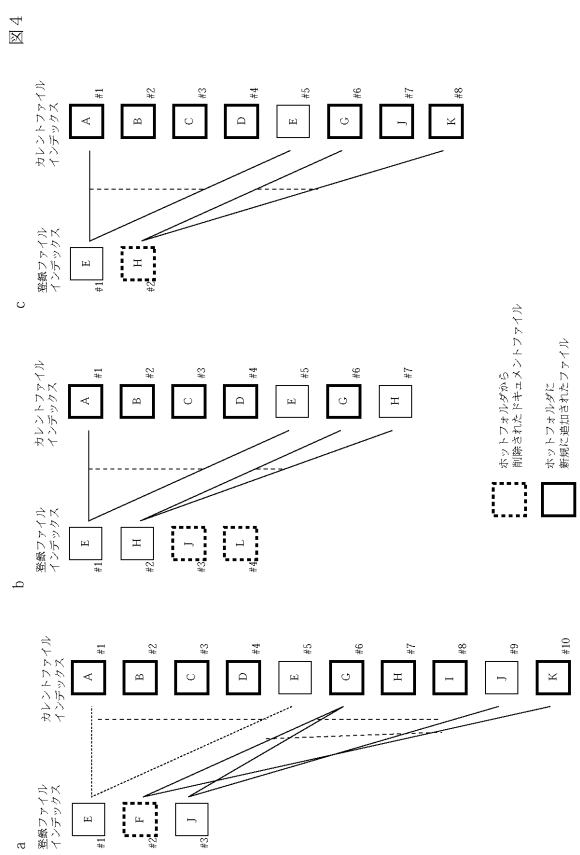
【 図 2 】



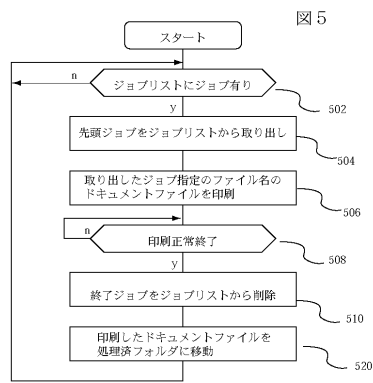
【 図 3 】



【 図 4 】



【 図 5 】



フロントページの続き

(58)調査した分野(Int.Cl. , D B名)

G06F 3/12